Identification of Nonlinear Systems Using Generalized Kernel Models

S. Chen, X. Hong, C.J. Harris and X.X. Wang

Abstract—**Nonlinear system identification is considered using a generalized kernel regression model. Unlike the standard kernel model, which employs a fixed common variance for all the kernel regressors, each kernel regressor in the generalized kernel model has an individually tuned diagonal covariance matrix that is determined by maximizing the correlation between the training data and the regressor using a repeated guided random search based on boosting optimization. An efficient construction algorithm based on orthogonal forward regression with leave-one-out test statistic and local regularization is then used to select a parsimonious generalized kernel regression model from the resulting full regression matrix. The proposed modeling algorithm is fully automatic and the user is not required to specify any criterion to terminate the construction procedure. Experimental results involving two real data sets demonstrate the effectiveness of the proposed nonlinear system identification approach.**

Keywords—**Nonlinear system identification, neural networks, regression, kernel model, orthogonal least squares, cross validation, leave-one-out test score, correlation.**

I. Introduction

Most systems encountered in the real world are nonlinear and in many practical applications nonlinear models are required to achieve an adequate modeling accuracy. A fundamental principle in system modeling is that the model should be no more complex than is required to capture the underlying system dynamics. This concept, known as the parsimonious principle, is particularly relevant in nonlinear model building because the size of a nonlinear model can easily become explosively large [1]. Forward selection using the orthogonal least squares (OLS) algorithm [2]–[10] is an effective construction method that is capable of producing parsimonious linear-in-the-weights nonlinear models with excellent generalization performance. Alternatively, the state-of-art sparse kernel modeling techniques, such as the support vector machine and relevant vector machine [11]–[19], have been gaining popularity in data modeling applications. These existing sparse regression modeling techniques typically place the kernel centers or mean vectors at the training input data and use a fixed common kernel variance for all the regressors. The value of this common kernel variance has a crucial influence on the sparsity level and generalization capability of the resulting model, and it has to be determined via cross validation. For example, in [5] a genetic algorithm is applied to determine the appropriate common kernel variance through optimizing the model generalization performance using a separate validation data set.

In this paper, we extend the standard kernel modeling approach. Specifically, we consider the use of a generalized kernel model for nonlinear systems, in which each kernel regressor has an individually tuned diagonal covariance matrix. Such a generalized kernel regression model has the potential of enhancing modeling capability and producing sparser final models, compared with the standard approach of single fixed common variance. The difficult issue however is how to determine these kernel covariance matrices. We note that the correlation function between a kernel regressor and the training data defines the "similarity" between the regressor and the training data and it can be used to "shape" the regressor by adjusting the associated kernel covariance matrix in order to maximize the absolute value of this correlation function. A guided random search method, referred to as the weighted optimization algorithm, is considered to perform the associated optimization task. This weighted optimization algorithm has its root from boosting [20]-[23]. Since the solution obtained by this weighted optimization algorithm may depend on the initial choice of population, the algorithm is augmented into a repeated weighted optimization method to provide a robust optimization and guarantee stable "global" solutions regardless the initial choice of population. The determination of kernel covariance matrices basically provides the pool of regressors or the full regression matrix, from which a parsimonious subset model can be selected using a standard kernel model construction approach.

The construction algorithm that we adopt to select a sparse generalized kernel model is the one that uses an OLS selection with the leave-one-out (LOO) test score and local regularization (LR) [10], which will be referred to as the LROLS with LOO score for short. The motivation of this construction algorithm is twofold. Firstly, the objective of modeling should be to optimize model generalization capability or test performance, rather than aiming to minimize the training mean square error (MSE). Moreover, it is highly desired that the model building process is automatic without the need for the user to specify some additional termination criterion. The so-called delete-one cross validation with its associated LOO score [8],[24]-[29] provides the capability to achieve this aim, without resorting to use a separate validation data set. Secondly, the computational efficiency and level of sparsity are crucial to the model construction process. The computational efficiency of adopting the LOO test score is ensured by using the OLS algorithm, as is shown in [8],[10], and multiple-regularizers or LR is known to be capable of providing very sparse solutions [6],[9],[10],[15]. The previous work [10] has shown that the LROLS with LOO score offers considerable advantages in realizing these two critical objectives of sparse modeling over several other state-of-art methods. The outline of the paper is as follows. Section II presents the generalized kernel regression model for nonlinear system identification.

S. Chen and C.J. Harris are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K., E-mails: {sqc,cjh}@ecs.soton.ac.uk; X. Hong is with Department of Cybernetics, University of Reading, Reading RG6 6AY, U.K., E-mail: x.hong@reading.ac.uk; X.X. Wang is with Department of Creative Technologies, University of Portsmouth, Portsmouth PO1 3HE, U.K., E-mail: xunxian.wang@port.ac.uk

Section III describes the proposed approach for the construction of sparse generalized kernel models. Section IV gives our modeling experiments, while Section V offers our conclusions.

II. Generalized kernel regression model

Consider a general discrete stochastic nonlinear system represented by [30]:

$$\begin{aligned} y_k &= f_s(y_{k-1}, \cdots, y_{k-n_y}, u_{k-1}, \cdots, u_{k-n_u}; \boldsymbol{\theta}) + \epsilon_k \\ &= f_s(\mathbf{x}_k; \boldsymbol{\theta}) + \epsilon_k \end{aligned} \quad (1)$$

where u_k and y_k are the system input and output variables, respectively, n_u and n_y are positive integers representing the known lags in u_k and y_k, respectively, the observation noise ϵ_k is uncorrelated with zero mean, $\mathbf{x}_k = [y_{k-1} \cdots y_{k-n_y}\ u_{k-1} \cdots u_{k-n_u}]^T$ denotes the system input vector with a known dimension $n = n_y + n_u$, $f_s(\bullet)$ is *a priori* unknown system mapping, and $\boldsymbol{\theta}$ is an unknown parameter vector associated with the appropriate, but yet to be determined, model structure. The system model (1) is to be identified from an N-sample system observational data set $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$, using some suitable functional which can approximate $f_s(\bullet)$ with arbitrary accuracy. One class of such functionals is the regression model of the form:

$$y_k = \hat{y}_k + \epsilon_k = \sum_{i=1}^{N_m} \theta_i g_i(\mathbf{x}_k) + \epsilon_k \quad (2)$$

where $\hat{y}_k$ denotes the model output given the input $\mathbf{x}_k$, θ_i are the model weight parameters, $g_i(\bullet)$ are the model regressors, and N_m is the total number of candidate regressors. The model (2) is very general and includes all the kernel based models, the polynomial-expansion model [2] and the general linear-in-the-weights nonlinear model [31]. In particular, for a kernel based model, the kernel mean vectors are placed at the training input data points giving rise to $N_m = N$, and the regressor $g_i(\mathbf{x})$ takes the form

$$g_i(\mathbf{x}) = \varphi\left(\sqrt{(\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)/\sigma^2}\right),\ 1 \le i \le N \quad (3)$$

where $\mathbf{x}_i$ are the training input vectors, σ^2 is a common kernel variance and $\varphi(\bullet)$ a chosen kernel function.

We will model the unknown dynamical process (1) by using a generalized kernel regression model. Specifically, we allow the kernel regressor $g_i(\mathbf{x})$ defined in (3) to be extended to:

$$g_i(\mathbf{x}) = \varphi\left(\sqrt{(\mathbf{x} - \mathbf{x}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \mathbf{x}_i)}\right) \quad (4)$$

where the ith kernel covariance matrix takes the form of $\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \cdots, \sigma_{i,n}^2\}$. For example, the generalized Gaussian kernel model adopts a general Gaussian function regressor $g_i(\mathbf{x}) = G(\mathbf{x}; \mathbf{x}_i, \boldsymbol{\Sigma}_i)$ with

$$G(\mathbf{x}; \mathbf{x}_i, \boldsymbol{\Sigma}_i) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \mathbf{x}_i)\right) \quad (5)$$

This generalized kernel model will have better modeling capability than the standard kernel model. However, it is more difficult to construct, as all the diagonal kernel covariance matrices must be specified.

With the regressor taking the form of (4), the regression model (2) becomes a generalized kernel model. This kernel model for the data point $(\mathbf{x}_k, y_k) \in D_N$ can be expressed as

$$y_k = \hat{y}_k + \epsilon_k = \mathbf{g}^T(k)\boldsymbol{\theta} + \epsilon_k \quad (6)$$

with the following notations

$$\boldsymbol{\theta} = [\theta_1\ \theta_2 \cdots \theta_N]^T \quad (7)$$

$$\mathbf{g}(k) = [g_1(\mathbf{x}_k)\ g_2(\mathbf{x}_k) \cdots g_N(\mathbf{x}_k)]^T \quad (8)$$

Furthermore, this generalized kernel model over the training set D_N can be written in the matrix form as

$$\mathbf{y} = \mathbf{G}\boldsymbol{\theta} + \boldsymbol{\epsilon} \quad (9)$$

by defining the following additional notations

$$\mathbf{y} = [y_1\ y_2 \cdots y_N]^T \quad (10)$$

$$\boldsymbol{\epsilon} = [\epsilon_1\ \epsilon_2 \cdots \epsilon_N]^T \quad (11)$$

$$\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2 \cdots \mathbf{g}_N] \quad (12)$$

$$\mathbf{g}_i = [g_i(\mathbf{x}_1)\ g_i(\mathbf{x}_2) \cdots g_i(\mathbf{x}_N)]^T,\ 1 \le i \le N \quad (13)$$

Note that $\mathbf{g}_k$ denotes the kth column of the regression matrix $\mathbf{G}$, while $\mathbf{g}(k)$ is the kth row of $\mathbf{G}$.

Let an orthogonal decomposition of the regression matrix $\mathbf{G}$ be

$$\mathbf{G} = \boldsymbol{\Phi}\mathbf{A} \quad (14)$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \quad (15)$$

and

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N] \quad (16)$$

with orthogonal columns that satisfy $\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j = 0$, if $i \ne j$. The regression model (9) can alternatively be expressed as

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{w} + \boldsymbol{\epsilon} \quad (17)$$

where the weight vector $\mathbf{w} = [w_1\ w_2 \cdots w_N]^T$, defined in the new space $\boldsymbol{\Phi}$, satisfy the triangular system

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{w} \quad (18)$$

Knowing $\mathbf{A}$ and $\mathbf{w}$, $\boldsymbol{\theta}$ can readily be solved from (18). The space spanned by the original model bases $\mathbf{g}_i$, $1 \le i \le N$, is identical to the space spanned by the orthogonal bases $\boldsymbol{\phi}_i$, $1 \le i \le N$, and the model output $\hat{y}_k$ is equivalently expressed by

$$\hat{y}_k = \boldsymbol{\phi}^T(k)\mathbf{w} \quad (19)$$

where $\boldsymbol{\phi}(k) = [\phi_{k,1}\ \phi_{k,2} \cdots \phi_{k,N}]^T$ is the kth row of $\boldsymbol{\Phi}$.

III. A CONSTRUCTION ALGORITHM FOR GENERALIZED KERNEL MODELS

The objective of sparse modeling is to construct a subset model consisting of N_s ($\ll N$) significant regressors only from the full set of regressors defined in (13), which can adequately model the underlying system (1).

A. Determination of the full regression matrix

To specify the pool of regressors or the full regression matrix $\mathbf{G}$, one needs to determine all the associated diagonal covariance matrices $\mathbf{\Sigma}_i, 1 \leq i \leq N$. The correlation between a regressor $\mathbf{g}_i$ and the training data is defined by

$$C(\mathbf{\Sigma}_i) = \frac{\mathbf{y}^T \mathbf{g}_i}{\sqrt{\mathbf{y}^T \mathbf{y}} \sqrt{\mathbf{g}_i^T \mathbf{g}_i}} \tag{20}$$

This correlation represents the "similarity" between $\mathbf{g}_i$ and $\mathbf{y}$, and it is a function of the regressor's kernel covariance matrix. Thus we can adopt this correlation function as the optimization criterion to determine the regressor's kernel covariance matrix. Specifically, we should choose $\mathbf{\Sigma}_i$ so that $|C(\mathbf{\Sigma}_i)|$ is maximized. We now explain why this is a good strategy to specify the pool of regressors. Let us first define the least squares cost or MSE associated with an m-term model as

$$S_m = \frac{1}{N} \sum_{k=1}^{N} (y_k - \hat{y}_k)^2 \tag{21}$$

where for the notational simplicity the same notation $\hat{y}_k$ is also used for representing the m-term model output. Obviously $S_0 = \mathbf{y}^T \mathbf{y}/N = \|\mathbf{y}\|^2/N$. Assuming that $\mathbf{g}_i$ is selected to form a one-term model, the associated reduction in the MSE value can be shown to be

$$\Delta S = S_0 - S_1 = \frac{\left(\mathbf{y}^T \mathbf{g}_i\right)^2}{\mathbf{g}_i^T \mathbf{g}_i} \tag{22}$$

which can be rewritten as

$$\Delta S = \left(\mathbf{y}^T \mathbf{y}\right) \frac{\left(\mathbf{y}^T \mathbf{g}_i\right)^2}{\left(\mathbf{y}^T \mathbf{y}\right)\left(\mathbf{g}_i^T \mathbf{g}_i\right)} = \|\mathbf{y}\|^2 \left|C(\mathbf{\Sigma}_i)\right|^2 \tag{23}$$

Since $\|\mathbf{y}\|^2$ is a constant, maximizing $|C(\mathbf{\Sigma}_i)|$ leads to a maximum reduction in the MSE value.

With the correlation function as the optimization criterion, we now turn our attention to optimization algorithm. We propose a repeated guided random search method to perform the associated optimization tasks. This method adopts ideas from boosting [20]-[23]. The basic component of the proposed optimizer is the weighted optimization algorithm, which is a simple guided random search method with boosting mechanism. Given the training data D_N and for fitting the lth regressor's covariance matrix, the algorithm is summarized as follows.

Weighted optimization algorithm

Initialization: Set iteration index $t = 0$, give the p randomly chosen initial values for $\mathbf{\Sigma}_l$, $\mathbf{\Sigma}_l^{(1)}(t), \mathbf{\Sigma}_l^{(2)}(t), \cdots, \mathbf{\Sigma}_l^{(p)}(t)$, with the associated weightings $\delta_i^{(t)} = \frac{1}{p}$ for $1 \leq i \leq p$, and specify a small positive value ξ for terminating the search.

Step 1: Boosting

1. Calculate the loss of each point in the population, namely

$$\text{cost}_i = 1 - |C(\mathbf{\Sigma}_l^{(i)}(t))|, \ 1 \leq i \leq p$$

2. Find

$$\mathbf{\Sigma}_l^{\text{best}}(t) = \arg\min\{\text{cost}_i, \ 1 \leq i \leq p\}$$

and

$$\mathbf{\Sigma}_l^{\text{worst}}(t) = \arg\max\{\text{cost}_i, \ 1 \leq i \leq p\}$$

3. Normalize the loss

$$\text{loss}_i = \frac{\text{cost}_i}{\sum_{j=1}^{p} \text{cost}_j}, \ 1 \leq i \leq p$$

4. Compute a weighting factor β_t according to

$$\eta_t = \sum_{i=1}^{p} \delta_i^{(t)} \text{loss}_i, \ \ \beta_t = \frac{\eta_t}{1 - \eta_t}$$

5. Update the weighting vector

$$\delta_i^{(t+1)} = \begin{cases} \delta_i^{(t)} \beta_t^{\text{loss}_i} & \text{for } \beta_t \leq 1, \\ \delta_i^{(t)} \beta_t^{1-\text{loss}_i} & \text{for } \beta_t > 1, \end{cases} \quad 1 \leq i \leq p$$

6. Normalize the weighting vector

$$\delta_i^{(t+1)} = \frac{\delta_i^{(t+1)}}{\sum_{j=1}^{p} \delta_j^{(t+1)}}, \ 1 \leq i \leq p$$

Step 2: Parameter updating

1. Construct the $(p+1)$th point using the formula

$$\mathbf{\Sigma}_l^{(p+1)}(t) = \sum_{i=1}^{p} \delta_i^{(t+1)} \mathbf{\Sigma}_l^{(i)}(t)$$

2. Construct the $(p+2)$th point using the formula

$$\mathbf{\Sigma}_l^{(p+2)}(t) = \mathbf{\Sigma}_l^{\text{best}}(t) + \left(\mathbf{\Sigma}_l^{\text{best}}(t) - \mathbf{\Sigma}_l^{(p+1)}(t)\right)$$

3. Choose a better point (smaller loss value) from $\mathbf{\Sigma}_l^{(p+1)}(t)$ and $\mathbf{\Sigma}_l^{(p+2)}(t)$ to replace $\mathbf{\Sigma}_l^{\text{worst}}(t)$, which will inherit the weighting δ value from $\mathbf{\Sigma}_l^{\text{worst}}(t)$.

Set $t = t + 1$ and repeat from *Step 1* until

$$\left\| \mathbf{\Sigma}_l^{(p+1)}(t) - \mathbf{\Sigma}_l^{(p+1)}(t-1) \right\| < \xi$$

Then choose the lth regressor covariance matrix as $\mathbf{\Sigma}_l = \mathbf{\Sigma}_l^{\text{best}}(t)$.

The algorithmic parameter that needs to be set appropriately is the population size p. The above weighted optimization algorithm performs a guided random search. However, the solution obtained may depend on the initial choice of population. To derive a robust algorithm that guarantees a "global" optimal solution, we augment the algorithm into the following repeated weighted optimization algorithm.

Repeated weighted optimization algorithm

Initialization: Give a positive integer number M for controlling the maximum repeating times, and choose a small positive number ξ_1 for terminating the search.

First generation: Randomly choose the p number of the initial population $\mathbf{\Sigma}_l^{(1)}, \cdots, \mathbf{\Sigma}_l^{(p)}$, and call the weighted optimization algorithm to obtain a solution $\mathbf{\Sigma}_l^{\text{best}}$.

Repeat loop: For $i = 1 : M$

Set $\mathbf{\Sigma}_l^{(1)} = \mathbf{\Sigma}_l^{\text{best}}$, and randomly generate the other $p-1$ points $\mathbf{\Sigma}_l^{(i)}$ for $2 \le i \le p$.

Call the weighted optimization algorithm to obtain a solution $\mathbf{\Sigma}_l^{\text{best}}$.

If $\left\| \mathbf{\Sigma}_l^{(1)} - \mathbf{\Sigma}_l^{\text{best}} \right\| < \xi_1$

Exit loop;

End if

End for

Choose the lth regressor's covariance matrix as $\mathbf{\Sigma}_l = \mathbf{\Sigma}_l^{\text{best}}$.

The important algorithmic parameters that need to be chosen appropriately are the maximum repeating times M and the termination criterion ξ_1. To further simplify control, we may simply let the loop repeat M times. Then we only needs to set an appropriate value for M. We have applied this repeated weighted optimization algorithm as a generic global optimizer in several difficult optimization applications [32], and analysis and empirical results given in [32] have shown that this guided random search algorithm is effective. The need to determine the diagonal covariance matrices of every candidate regressors represents additional computational complexity of the proposed generalized kernel modeling approach, in comparison with the standard kernel method. However, the standard kernel approach would typically require cross validation for specifying the common single kernel variance, and this may involve additional validation data set and can also be computationally expensive. The proposed method does not require cross validation to tune kernel parameters, which is an important practical advantage.

B. The LROLS algorithm with LOO test score for subset model selection

Once the full regression matrix $\mathbf{G}$ has been designed, the LROLS algorithm with the LOO test score [10] can be used to select a subset model. In this construction algorithm, the weight parameter vector $\mathbf{w}$ is the regularized least squares solution obtained by minimizing the following regularized error criterion

$$J_R(\mathbf{w}, \boldsymbol{\lambda}) = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} + \sum_{i=1}^{N} \lambda_i w_i^2 \tag{24}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \cdots \lambda_N]^T$ is the regularization parameter vector, which is optimized based on the evidence procedure [33] with the iterative updating formulas [9],[10]

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}} \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{w_i^2}, \ \ 1 \le i \le N \tag{25}$$

where

$$\gamma_i = \frac{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i}{\lambda_i + \boldsymbol{\phi}_i^T \boldsymbol{\phi}_i} \quad \text{and} \quad \gamma = \sum_{i=1}^{N} \gamma_i \tag{26}$$

Usually a few iterations (typically less than 10) are sufficient to find a local optimal $\boldsymbol{\lambda}$. The criterion (24) has its root in the Bayesian learning framework. This Bayesian interpretation of $J_R(\mathbf{w}, \boldsymbol{\lambda})$ together with the full derivation of the updating formulas (25) and (26) can be found in [9].

A forward selection procedure is used to construct a sparse model by incrementally minimizing the LOO test score. Assume that an m-term model is selected from the full model (17). Then the LOO test error [24],[27]-[29], denoted as $\epsilon_k^{(m,-k)}$, for the selected m-term model can be shown to be [8],[10]

$$\epsilon_k^{(m,-k)} = \frac{\epsilon_k^{(m)}}{\eta_k^{(m)}} \tag{27}$$

where $\epsilon_k^{(m)}$ is the m-term modeling error and $\eta_k^{(m)}$ is the associated LOO error weighting given by

$$\eta_k^{(m)} = 1 - \sum_{i=1}^{m} \frac{\phi_{k,i}^2}{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i + \lambda_i} \tag{28}$$

The mean square LOO error for the model with a size m is defined by

$$J_m = E\left[\left(\epsilon_k^{(m,-k)} \right)^2 \right] = \frac{1}{N} \sum_{k=1}^{N} \frac{\left(\epsilon_k^{(m)} \right)^2}{\left(\eta_k^{(m)} \right)^2} \tag{29}$$

This LOO test score is a measure of the model generalization performance and it can be computed efficiently due to the fact that the m-term model error $\epsilon_k^{(m)}$ and the associated LOO error weighting $\eta_k^{(m)}$ can be calculated recursively according to

$$\epsilon_k^{(m)} = y_k - \sum_{i=1}^{m} \phi_{k,i} w_i = \epsilon_k^{(m-1)} - \phi_{k,m} w_m \tag{30}$$

and

$$\begin{aligned} \eta_k^{(m)} &= 1 - \sum_{i=1}^{m} \frac{\phi_{k,i}^2}{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i + \lambda_i} \\ &= \eta_k^{(m-1)} - \frac{\phi_{k,m}^2}{\boldsymbol{\phi}_m^T \boldsymbol{\phi}_m + \lambda_m} \end{aligned} \tag{31}$$

respectively. For the benefits of those readers who are unfamiliar with the LOO statistics, the idea of delete-one cross validation and the computation of the LOO test error are explained in Appendix A.

The subset model selection procedure can be carried as follows: at the mth stage of the selection procedure, a model term is selected among the remaining m to N candidates if the resulting m-term model produces the smallest LOO test score J_m. It has been shown in [8] that the LOO statistic J_m is convex with respect to the model size m. That is, there exists an "optimal" model size N_s such that for $m \le N_s$ J_m decreases as m increases while for $n \ge N_s + 1$ J_m increases as m increases. This property is extremely useful, as it enables the selection procedure to be automatically terminated with an N_s-term model when $J_{N_s+1} > J_{N_s}$, without the need for

the user to specify a separate termination criterion. The iterative procedure for constructing a sparse generalized kernel model based on the LROLS with the LOO test score can now be summarized:

Initialization. Set $\lambda_i, 1 \leq i \leq N$, to the same small positive value (e.g. 0.0001). Set iteration index $I = 1$.

Step 1. Given the current $\boldsymbol{\lambda}$ and with the following initial conditions

$$\begin{aligned} &\epsilon_k^{(0)} = y_k \text{ and } \eta_k^{(0)} = 1,\ 1 \leq k \leq N \\ &J_0 = \tfrac{1}{N}\mathbf{y}^T\mathbf{y} = \tfrac{1}{N}\sum_{k=1}^{N} y_k^2 \end{aligned} \tag{32}$$

use the procedure described in Appendix B to select a subset model with N_I terms.

Step 2. Update $\boldsymbol{\lambda}$ using (25) and (26) with $N = N_I$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a pre-set maximum iteration number (e.g. 10) is reached, stop; otherwise set $I+ = 1$ and go to *Step 1*.

The computational complexity of the above algorithm is dominated by the 1st iteration. After the 1st iteration, the model set contains only $N_1(\ll N)$ terms, and the complexity of the subsequent iteration decreases dramatically. It is worth emphasizing that regressor selection is based on the LOO statistic, not the usual training MSE. Thus, the subset model selection is directly based on the model generalization capability using a single training set, with the local regularization further enforcing sparsity. Moreover, the subset model selection is fully automatic, and the user does not require to specify a termination criterion.

IV. Modeling examples

Two real-data sets were used to demonstrate the effectiveness of the proposed approach for constructing sparse generalized kernel models. The population size p and the maximum repeating times M for fitting kernel covariance matrices were chosen empirically to ensure that the subset selection procedure could produce consistent final models with the same levels of modeling accuracy and model sparsity for repeating runs. Empirically, it was found that the values of p and M did not critically influence the modeling result.

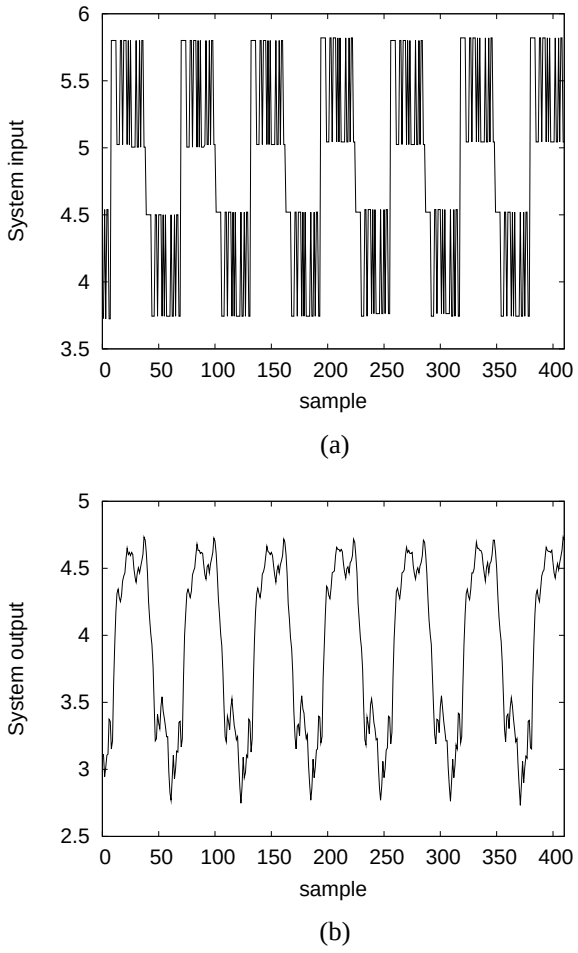


Fig. 1. The engine data set: (a) system input u_k, and (b) system output y_k.

TABLE I

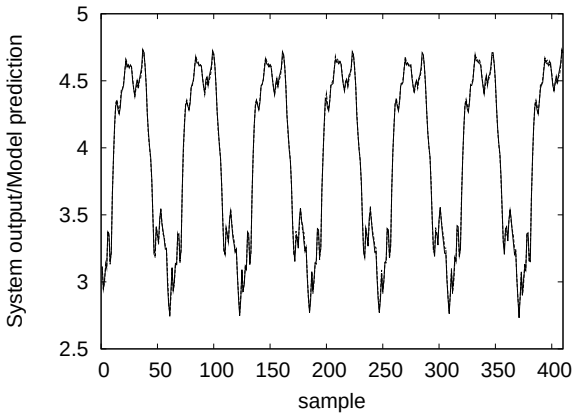
Subset generalized Gaussian kernel model generated for the engine data set by the LROLS algorithm with the LOO test score. The kernel covariance matrices are determined by maximizing the correlation criterion using the repeated weighted optimization algorithm.

term l	mean vector $\mathbf{x}_l$			diagonal covariance $\boldsymbol{\Sigma}_l$			weight θ_l
1	4.72520e+0	5.02450e+0	5.80060e+0	4.87424e+0	1.32234e+2	1.37096e+1	-6.82324e+1
2	4.61830e+0	5.00510e+0	5.80060e+0	4.24816e+0	2.10933e+2	1.42741e+1	2.14508e+2
3	4.59540e+0	5.82000e+0	5.82000e+0	3.70595e+0	1.87710e+1	3.87729e+2	-5.91563e+1
4	4.51140e+0	5.00510e+0	5.80060e+0	3.56184e+0	1.37055e+2	1.49309e+1	-1.47935e+2
5	3.11380e+0	4.53940e+0	4.53940e+0	4.00000e+2	4.00000e+2	4.00000e+2	-2.21061e+2
6	4.16770e+0	5.80060e+0	5.80060e+0	1.76914e+0	7.20294e+1	3.54250e+2	3.90802e+1
7	4.39680e+0	5.80060e+0	5.02450e+0	3.12161e+2	7.14241e+0	4.67573e+0	4.30384e+0
8	4.55720e+0	5.80060e+0	5.00510e+0	4.72670e+0	1.20541e+1	1.53883e+1	-2.08342e+1
9	2.86180e+0	3.74390e+0	4.52000e+0	4.00000e+2	4.00000e+2	4.00000e+2	2.15294e+2
10	4.63360e+0	5.80060e+0	5.00510e+0	4.03028e+0	1.56167e+1	9.56584e+1	8.30105e+1
11	4.12190e+0	4.50060e+0	4.50060e+0	1.63687e+0	3.34953e+2	1.12488e+2	-2.44486e+1
12	4.61830e+0	5.02450e+0	5.80060e+0	3.50205e+0	1.12860e+1	2.81422e+2	-9.63619e+0
13	3.16730e+0	5.80060e+0	3.74390e+0	3.62482e+2	5.98815e+0	2.43691e+2	4.47388e+0
14	4.39680e+0	5.80060e+0	5.00510e+0	3.86510e+0	7.20823e+0	7.31132e+1	-1.11560e+1
15	4.31280e+0	5.00510e+0	5.00510e+0	2.27494e+0	9.24282e+0	2.39816e+2	7.25891e+0

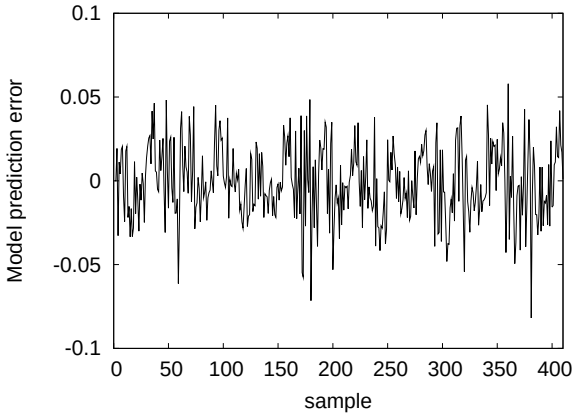
TABLE II

SUBSET GENERALIZED GAUSSIAN KERNEL MODEL GENERATED FOR THE GAS FURNACE DATA SET BY THE LROLS ALGORITHM WITH THE LOO TEST SCORE. THE KERNEL COVARIANCE MATRICES ARE DETERMINED BY MAXIMIZING THE CORRELATION CRITERION USING THE REPEATED WEIGHTED OPTIMIZATION ALGORITHM.

term l	mean vector $\mathbf{x}_l$ diagonal covariance $\mathbf{\Sigma}_l$						weight θ_l
1	5.80000e+1	5.56000e+1	5.35000e+1	-2.33000e+0	-2.47300e+0	-2.49900e+0	-7.13848e+1
	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	
2	5.54000e+1	5.52000e+1	5.53000e+1	-1.14900e+0	-1.61900e+0	-1.62800e+0	7.74589e+1
	4.00000e+2	4.00000e+2	4.00000e+2	1.70286e+2	1.75048e+2	8.44736e+1	
3	5.71000e+1	5.65000e+1	5.60000e+1	-7.00000e-3	-1.60000e-1	-4.88000e-1	-2.79416e+2
	3.30424e+2	3.30759e+2	3.81122e+2	4.91483e+1	4.00000e+2	3.23049e+2	
4	5.85000e+1	5.86000e+1	5.80000e+1	1.95000e-1	2.53000e-1	2.04000e-1	6.14813e+2
	3.99072e+2	3.99094e+2	3.99603e+2	1.45628e+2	7.63318e+1	8.58759e+1	
5	5.04000e+1	5.02000e+1	5.04000e+1	-6.03000e-1	-5.53000e-1	-1.61000e-1	-6.37936e+1
	3.96396e+2	3.96400e+2	3.95757e+2	1.86644e+1	3.92835e+2	2.40638e+2	
6	5.73000e+1	5.78000e+1	5.83000e+1	-1.82000e-1	1.70000e-2	1.31000e-1	-5.59610e+2
	3.96126e+2	3.90090e+2	3.91998e+2	2.66949e+2	5.14039e+1	7.70242e+1	
7	5.24000e+1	5.20000e+1	5.20000e+1	-1.05500e+0	-5.88000e-1	-1.80000e-1	-2.41017e+1
	3.31951e+2	4.00000e+2	3.95641e+2	2.96358e+1	2.00516e+2	6.99495e+1	
8	5.43000e+1	5.30000e+1	5.26000e+1	-5.28000e-1	-7.40000e-1	-8.24000e-1	3.25157e+2
	3.97725e+2	3.98725e+2	3.96916e+2	3.37194e+1	3.96999e+2	3.96236e+2	
9	5.70000e+1	5.60000e+1	5.43000e+1	3.40000e-2	-2.04000e-1	-5.28000e-1	-5.50658e+2
	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	
10	5.86000e+1	5.80000e+1	5.70000e+1	2.53000e-1	2.04000e-1	3.40000e-2	-3.76651e+1
	4.00000e+2	4.00000e+2	2.97354e+2	1.84745e+2	4.20406e+1	3.26347e+2	
11	5.03000e+1	4.97000e+1	4.93000e+1	-1.55100e+0	-1.08000e+0	-2.80000e-1	-1.77366e+2
	4.00000e+2	4.00000e+2	4.00000e+2	4.91252e+1	3.30374e+1	4.00000e+2	
12	5.40000e+1	5.30000e+1	5.24000e+1	-1.52000e+0	-1.42100e+0	-1.05500e+0	1.11535e+2
	3.75376e+2	3.93381e+2	3.82803e+2	2.40545e+2	4.74020e+1	3.67877e+2	
13	4.97000e+1	4.93000e+1	4.92000e+1	-1.08000e+0	-2.80000e-1	2.55000e-1	5.93483e+1
	3.94282e+2	3.94917e+2	3.96847e+2	2.53283e+1	1.82365e+2	4.27622e+1	
14	5.80000e+1	5.70000e+1	5.60000e+1	2.04000e-1	3.40000e-2	-2.04000e-1	4.46719e+2
	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	
15	5.40000e+1	5.30000e+1	5.18000e+1	3.30000e-2	-6.76000e-1	-1.17500e+0	1.99946e+1
	3.08446e+2	3.91400e+2	3.91666e+2	3.90699e+2	3.83084e+2	3.76996e+1	
16	5.20000e+1	5.00000e+1	5.00000e+1	1.03200e+0	9.22000e-1	3.82000e-1	3.27588e+2
	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	
17	4.87000e+1	4.85000e+1	4.88000e+1	5.77000e-1	5.77000e-1	6.48000e-1	-1.73100e+2
	3.84430e+2	3.87019e+2	2.98609e+2	8.10929e+1	3.60865e+2	3.67886e+2	
18	4.97000e+1	4.93000e+1	4.94000e+1	-1.09900e+0	-7.14000e-1	-2.37000e-1	5.04839e+1
	4.00000e+2	3.99841e+2	4.00000e+2	4.00000e+2	1.46008e+1	4.00000e+2	
19	4.94000e+1	4.81000e+1	4.72000e+1	6.71000e-1	1.64000e-1	9.00000e-3	-5.81335e+1
	3.78884e+2	3.92997e+2	3.96774e+2	2.96011e+2	9.75484e+1	3.48400e+1	
20	4.93000e+1	4.94000e+1	5.00000e+1	-7.14000e-1	-2.37000e-1	2.18000e-1	9.82482e+1
	4.00000e+2	4.00000e+2	4.00000e+2	4.00000e+2	2.69222e+1	4.00000e+2	
21	5.00000e+1	5.00000e+1	5.04000e+1	9.22000e-1	3.82000e-1	2.50000e-2	-9.01877e+1
	2.79438e+2	4.00000e+2	4.00000e+2	2.41774e+2	1.42912e+2	4.00000e+2	

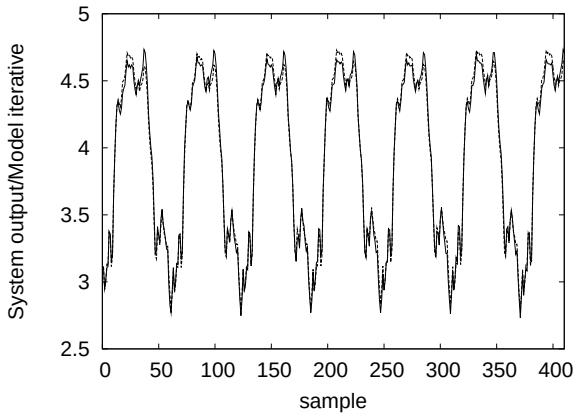


(a)

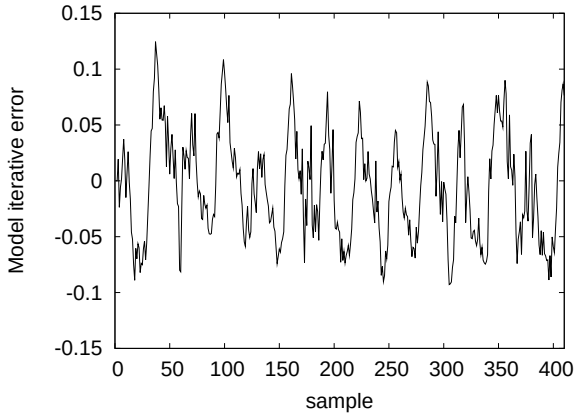


(b)

Fig. 2. Performance of the 15-term generalized Gaussian kernel model for the engine data set: (a) the model prediction $\hat{y}_k$ (dashed) superimposed on the system output y_k (solid), and (b) the model prediction error $\epsilon_k = y_k - \hat{y}_k$.



(a)



(b)

Fig. 3. Performance of the 15-term generalized Gaussian kernel model for the engine data set: (a) the iterative model output $\hat{y}_{d,k}$ (dashed) superimposed on the system output y_k (solid), and (b) the iterative model error $\epsilon_{d,k} = y_k - \hat{y}_{d,k}$.

Example 1. This example constructed a model representing the relationship between the fuel rack position (input u_k) and the engine speed (output y_k) for a Leyland TL11 turbocharged, direct injection diesel engine operated at a low engine speed. Detailed system description and experimental setup can be found in [34]. The data set, depicted in Fig. 1, contained 410 samples. The first 210 data points were used in training and the last 200 points in model validation. The previous study [34] has shown that this data set can be modeled adequately by a nonlinear model of the form:

$$y_k = f_s(\mathbf{x}_k) + \epsilon_k \tag{33}$$

with $f_s(\bullet)$ describing the unknown underlying system and the system input vector defining by

$$\mathbf{x}_k = [y_{k-1}\ u_{k-1}\ u_{k-2}]^T \tag{34}$$

Since every training input data points were considered as a candidate regressor's center, there were $N = 210$ regressors for the full regression model. The previous results [9],[10] had shown that when fitting a Gaussian kernel model with a single common variance, $\sigma^2 = 1.69$ was the optimal value for this kernel variance. Various kernel modeling techniques were employed in [10] to fit this data set, and the best Gaussian kernel model was provided by the LROLS with the LOO test score, which consisted of 22 terms. The MSE values of this model over the training and validation sets were 0.000453 and 0.000490, respectively.

The proposed sparse model construction algorithm was applied to construct a generalized Gaussian kernel model for this data set. The kernel covariance matrices were first identified by optimizing the associated correlation criteria using the repeated weighted optimization algorithm with $p = 21$ and $M = 10$. The LROLS algorithm based on the LOO test score then selected a 15-term subset generalized Gaussian kernel model from the resulting full regression matrix, and the constructed model is given in Table I. The MSE values of this model were 0.000482 over the training set and 0.000496 over the validation set, respectively. The model prediction $\hat{y}_k$ and prediction error $\epsilon_k = y_k - \hat{y}_k$ generated by this model are illustrated in Fig. 2. The obtained 15-term generalized Gaussian kernel model was used to iteratively generate the model output according to

$$\hat{y}_{d,k} = f_m(\hat{\mathbf{x}}_{d,k}) \tag{35}$$

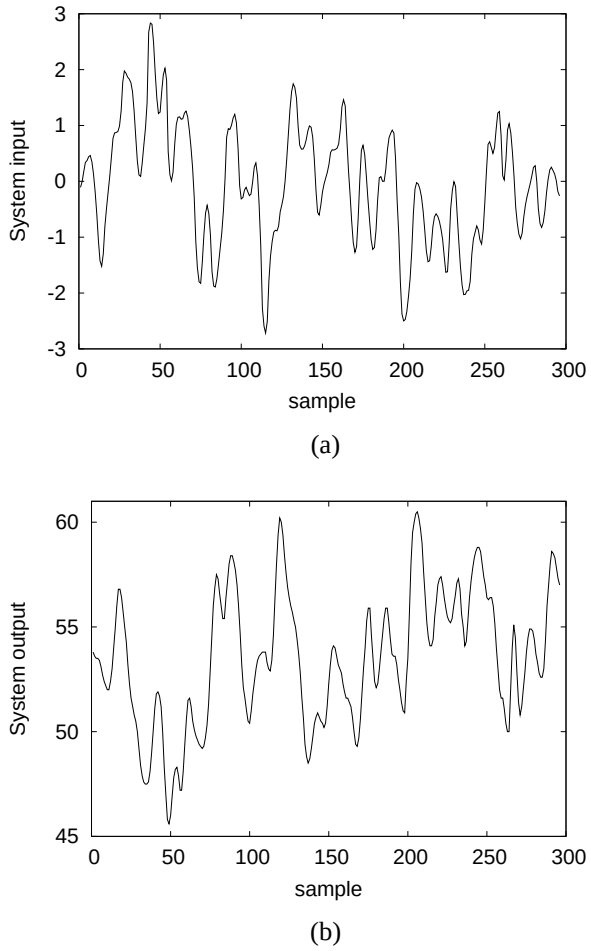


Fig. 4. The gas furnace data set: (a) system input u_k, and (b) system output y_k.

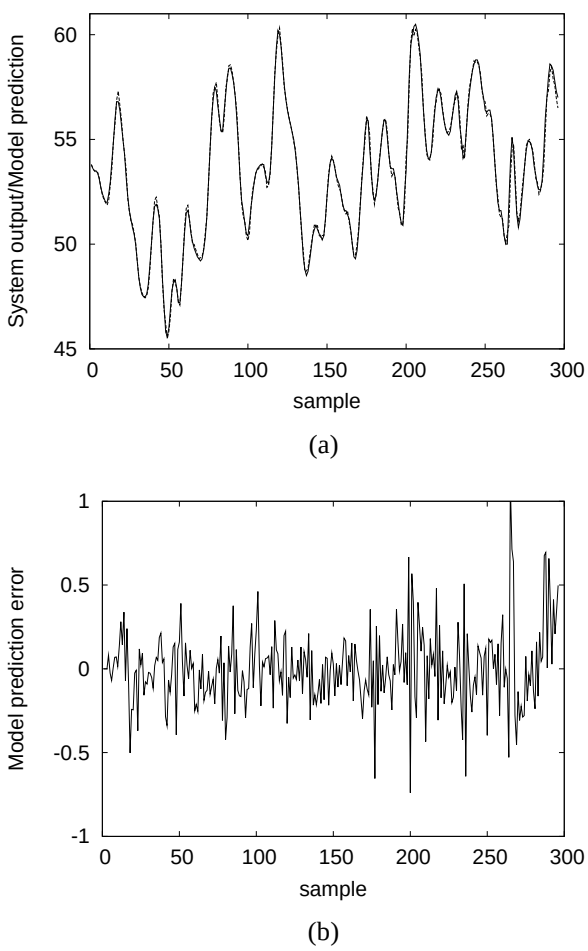


Fig. 5. Performance of the 21-term generalized Gaussian kernel model for the gas furnace data set: (a) the model prediction $\hat{y}_k$ (dashed) superimposed on the system output y_k (solid), and (b) the model prediction error $\epsilon_k = y_k - \hat{y}_k$.

with

$$\hat{\mathbf{x}}_{d,k} = [\hat{y}_{d,k-1}\ u_{k-1}\ u_{k-2}]^T \tag{36}$$

where $f_m(\bullet)$ denotes the model mapping. The iterative model output $\hat{y}_{d,k}$ and the iterative model error $\epsilon_{d,k} = y_k - \hat{y}_{d,k}$, are depicted in Fig. 3. Compared with the standard kernel method, the proposed generalized kernel modeling approach is able to produce more parsimonious model with a similar modeling accuracy.

Example 2. This example constructed a model for the gas furnace data set (Series J in [35]). The data set, illustrated in Fig. 4, contained 296 pairs of input-output points, where the input u_k was the coded input gas feed rate and the output y_k represented CO_2 concentration from the gas furnace. All the 296 data points were used in training, with the model input vector defined by

$$\mathbf{x}_k = [y_{k-1}\ y_{k-2}\ y_{k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}]^T \tag{37}$$

The number of candidate regressors was $N = 296$ for this data set. The previous experiments had found out that the existing state-of-art kernel regression techniques failed to fit a Gaussian kernel regression model using a common kernel variance [10]. Various existing kernel regression techniques were then used in [10] to fit a thin-plate-spline regression model for this data set, where the regressors were given by

$$g_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|^2 \log(\|\mathbf{x} - \mathbf{x}_i\|), \quad 1 \le i \le N \tag{38}$$

and the best result obtained was again given by the LROLS with the LOO test score, which yielded a 28-term thin-plate-spline model with a training MSE of 0.053306.

By adopting a generalized Gaussian kernel model structure, the LROLS with the LOO test score was able to identify a 21-term model, as listed in Table II, with a training MSE of 0.053452. The candidate regressors' kernel covariance matrices were fitted by optimizing the correlation criterion using the repeated weighted optimization with $p = 21$ and $M = 10$. The model prediction and prediction error generated by this 21-term generalized Gaussian kernel model are shown in Fig. 5. The obtained model was also used to iteratively produce the model output $\hat{y}_{d,k} = f_m(\hat{\mathbf{x}}_{d,k})$ given the input

$$\hat{\mathbf{x}}_{d,k} = [\hat{y}_{d,k-1}\ \hat{y}_{d,k-2}\ \hat{y}_{d,k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}]^T \tag{39}$$

The iterative model output and the associated modeling error $\epsilon_{d,k} = y_k - \hat{y}_{d,k}$ are illustrated in Fig. 6.

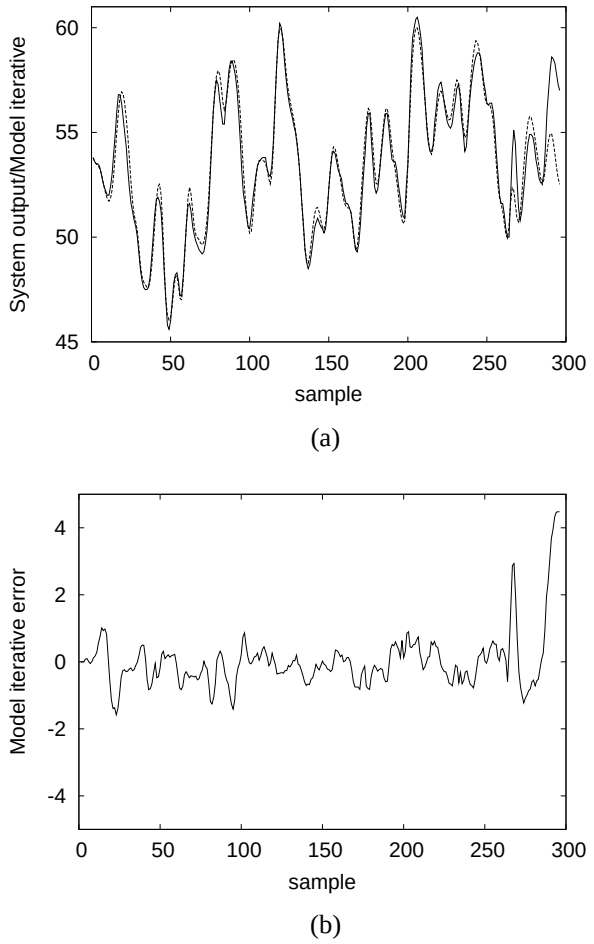


Fig. 6. Performance of the 21-term generalized Gaussian kernel model for the gas furnace data set: (a) the iterative model output $\hat{y}_{d,k}$ (dashed) superimposed on the system output y_k (solid), and (b) the iterative model error $\epsilon_{d,k} = y_k - \hat{y}_{d,k}$.

V. Conclusions

Identification of discrete-time nonlinear systems has been considered using a generalized kernel regression model structure. As with the standard kernel model, the kernel mean vectors are directly placed on the training input points. However, each regressor in the generalized kernel model has an individually fitted diagonal covariance matrix. This generalized kernel model structure thus has an enhanced modeling capability and is capable of producing more parsimonious models for nonlinear systems, compared with the standard kernel model structure. The design of the pool of regressors or the determination of the candidate kernel covariance matrices is performed by maximizing a correlation criterion using a repeated guided random search based on boosting optimization. The efficient OLS algorithm based on the leave-one-out test statistic and local regularization can then automatically select a sparse model from the resulting pool of candidate regressors. The effectiveness of the proposed nonlinear system identification approach has been demonstrated by the experimental results involving two real data sets.

Appendix A

Consider the model selection problem where a set of m models have been identified using the training data set D_N. Denote these models, identified using all the N data points of D_N, as $\hat{y}_k^{(j)}$ and the corresponding modeling errors as

$$\epsilon_k^{(j)} = y_k - \hat{y}_k^{(j)} \tag{40}$$

with index $j = 1, 2, \cdots, m$. A commonly used cross validation for model selection is the delete-one cross validation. The idea is as follows. For every model, each data point in the training set D_N is sequentially set aside in turn, a model is estimated using the remaining $N-1$ data points, and the prediction error is derived using only the data point that was removed from training. Specifically, let $D_{N,-l}$ be the resulting data set by removing the lth data point from D_N, and denote the jth model estimated using $D_{N,-l}$ as $\hat{y}_k^{(j,-l)}$ and the related predicted model residual at l as

$$\epsilon_l^{(j,-l)} = y_l - \hat{y}_l^{(j,-l)} \tag{41}$$

The mean square LOO test error [24],[27] for the jth model $\hat{y}_k^{(j)}$ is obtained by averaging all these prediction errors:

$$E\left[\left(\epsilon_k^{(j,-k)}\right)^2\right] = \frac{1}{N}\sum_{k=1}^{N}\left(\epsilon_k^{(j,-k)}\right)^2 \tag{42}$$

The mean square LOO test error is a measure of the model generalization capability. To select the best model from the m candidate models $\hat{y}_k^{(j)}$, $1 \le j \le m$, the same modeling procedure is applied to each of the m predictors, and the model with the minimum LOO test error is selected.

For linear-in-the-weights models, the LOO test errors can be generated, without actually sequentially splitting the training data set and repeatedly estimating the associated models, by using the Sherman-Morrison-Woodbury theorem [24]. Moreover within the forward model selection procedure using the OLS algorithm, the LOO test errors for the m-term model can be computed very efficiently. It can readily be shown [8],[10] that the computation of the LOO error $\epsilon_k^{(m,-k)}$ for the m-term model is based on the previously selected $(m-1)$-term model and the currently selected mth model term via the efficient recursion formulas (30) and (31).

Appendix B

The modified Gram-Schmidt orthogonalization procedure [2] calculates the $\mathbf{A}$ matrix row by row and orthogonalizes $\mathbf{G}$ as follows: at the lth stage make the columns $\mathbf{g}_j$, $l+1 \le j \le N$, orthogonal to the lth column and repeat the operation for $1 \le l \le N-1$. Specifically, denoting $\mathbf{g}_j^{(0)} = \mathbf{g}_j$, $1 \le j \le N$, then for $l = 1, 2, \cdots, N-1$,

$$\left.\begin{aligned} \boldsymbol{\phi}_l &= \mathbf{g}_l^{(l-1)} \\ a_{l,j} &= \boldsymbol{\phi}_l^T \mathbf{g}_j^{(l-1)} / \left(\boldsymbol{\phi}_l^T \boldsymbol{\phi}_l\right), \;\; l+1 \le j \le N \\ \mathbf{g}_j^{(l)} &= \mathbf{g}_j^{(l-1)} - a_{l,j}\boldsymbol{\phi}_l, \;\; l+1 \le j \le N \end{aligned}\right\} \tag{43}$$

The last stage of the procedure is simply $\boldsymbol{\phi}_N = \mathbf{g}_N^{(N-1)}$. The elements of $\mathbf{w}$ are computed by transforming $\mathbf{y}^{(0)} = \mathbf{y}$ in a similar way:

$$\left.\begin{aligned} w_l &= \boldsymbol{\phi}_l^T \mathbf{y}^{(l-1)} / \left(\boldsymbol{\phi}_l^T \boldsymbol{\phi}_l + \lambda_l\right) \\ \mathbf{y}^{(l)} &= \mathbf{y}^{(l-1)} - w_l \boldsymbol{\phi}_l \end{aligned}\right\} 1 \le l \le N \qquad (44)$$

This orthogonalization scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner [2]. First define

$$\mathbf{G}^{(l-1)} = \left[\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{l-1}\ \mathbf{g}_l^{(l-1)} \cdots \mathbf{g}_N^{(l-1)}\right] \qquad (45)$$

If some of the columns $\mathbf{g}_l^{(l-1)}, \cdots, \mathbf{g}_N^{(l-1)}$ in $\mathbf{G}^{(l-1)}$ have been interchanged, this will still be referred to as $\mathbf{G}^{(l-1)}$ for notational convenience. Let a very small positive number T_z be given, which specifies the zero threshold and is used to automatically avoiding any ill-conditioning or singular problem. With the initial conditions as specified in (32), the lth stage of the selection procedure is given as follows.

Step 1. For $l \le j \le N$:

Test–Conditioning number check. If $\left(\mathbf{g}_j^{(l-1)}\right)^T \mathbf{g}_j^{(l-1)} < T_z$, the jth candidate is not considered.
Compute

$$w_{l,j} = \left(\mathbf{g}_j^{(l-1)}\right)^T \mathbf{y}^{(l-1)} / \left(\left(\mathbf{g}_j^{(l-1)}\right)^T \mathbf{g}_j^{(l-1)} + \lambda_j\right)$$

and calculate, for $1 \le k \le N$,

$$\left.\begin{aligned} \epsilon_{k,j}^{(l)} &= y_k^{(l-1)} - g_{k,j}^{(l-1)} w_{l,j} \\ \eta_{k,j}^{(l)} &= \eta_k^{(l-1)} - \frac{\left(g_{k,j}^{(l-1)}\right)^2}{\left(\mathbf{g}_j^{(l-1)}\right)^T \mathbf{g}_j^{(l-1)} + \lambda_j} \end{aligned}\right\}$$

and

$$J_{l,j} = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{\epsilon_{k,j}^{(l)}}{\eta_{k,j}^{(l)}}\right)^2$$

where $y_k^{(l-1)}$ and $g_{k,j}^{(l-1)}$ are the kth elements of $\mathbf{y}^{(l-1)}$ and $\mathbf{g}_j^{(l-1)}$, respectively. Let the index set $\mathcal{J}_l$ be

$$\mathcal{J}_l = \{l \le j \le N \text{ and } j \text{ passes } \textbf{Test}\}$$

Step 2. Find

$$J_l = J_{l,j_l} = \min\{J_{l,j},\ j \in \mathcal{J}_l\}$$

Then the j_lth column of $\mathbf{G}^{(l-1)}$ is interchanged with the lth column of $\mathbf{G}^{(l-1)}$, the j_lth column of $\mathbf{A}$ is interchanged with the lth column of $\mathbf{A}$ up to the $(l-1)$th row, and the j_lth element of $\boldsymbol{\lambda}$ is interchanged with the lth element of $\boldsymbol{\lambda}$. This effectively selects the j_lth candidate as the lth regressor in the subset model.

Step 3. The selection procedure is terminated with a $(l-1)$-term model, if $J_l > J_{l-1}$. Otherwise, perform the orthogonalization as indicated in (43) to derive the lth row of $\mathbf{A}$ and to transform $\mathbf{G}^{(l-1)}$ into $\mathbf{G}^{(l)}$; calculate w_l and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (44); update the LOO error weightings

$$\eta_k^{(l)} = \eta_k^{(l-1)} - \frac{\phi_{k,l}^2}{\boldsymbol{\phi}_l^T \boldsymbol{\phi}_l + \lambda_l},\ \ k = 1, 2, \cdots, N$$

and go to *Step 1*.

REFERENCES

[1] S.A. Billings and S. Chen, "The determination of multivariable nonlinear models for dynamic systems," in: C.T. Leondes, ed., *Control and Dynamic Systems*, Volume 7 of *Neural Network Systems Techniques and Applications*. San Diego: Academic Press, 1998, pp.231–278.

[2] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, Vol.50, No.5, pp.1873–1896, 1989.

[3] S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.2, No.2, pp.302–309, 1991.

[4] S. Chen, E.S. Chng and K. Alkadhimi, "Regularised orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Control*, Vol.64, No.5, pp.829–837, 1996.

[5] S. Chen, Y. Wu and B.L. Luk, "Combined genetic algorithm optimisation and regularised orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.10, No.5, pp.1239–1243, 1999.

[6] S. Chen, "Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models," in *Proc. 6th Int. Conf. Signal Processing* (Beijing, China), Aug.26-30, 2002, Vol.2, pp.1229–1232.

[7] X. Hong and C.J. Harris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Trans. Neural Networks*, Vol.13, No.5, pp.1245–1250, 2002.

[8] X. Hong, P.M. Sharkey and K. Warwick, "Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic," *IEE Proc. Control Theory and Applications*, Vol.150, No.3, pp.245–254, 2003.

[9] S. Chen, X. Hong and C.J. Harris, "Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, Vol.48, No.6, pp.1029–1036, 2003.

[10] S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.34, No.2, pp.898–911, 2004.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[12] V. Vapnik, S. Golowich and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in: M.C. Mozer, M.I. Jordan and T. Petsche, eds., *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997, pp.281–287.

[13] P.M.L. Drezet and R.F. Harrison, "Support vector machines for system identification," in *Proc. UKACC Int. Conf. Control'98* (Swansea, U.K.), Sept.1-4, 1998, pp.688–692.

[14] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press, 2000.

[15] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, Vol.1, pp.211–244, 2001.

[16] B. Schlkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.

[17] K.L. Lee and S.A. Billings, "Time series prediction using support vector machines, the orthogonal and the regularized orthogonal least-squares algorithms," *Int. J. Systems Science*, Vol.33, No.10, pp.811–821, 2002.

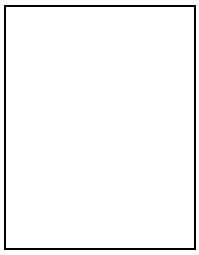
[18] L. Zhang, W. Zhou and L. Jiao, "Wavelet support vector machine," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.34, No.1, pp.34–39, 2004.

[19] W. Chu, S.S. Keerthi and C.J. Ong, "Bayesian support vector regression using a unified loss function," *IEEE Trans. Neural Networks*, Vol.15, No.1, pp.29–44, 2004.

[20] R.E. Schapire, "The strength of weak learnability," *Machine Learning*, Vol.5, No.2, pp.197–227, 1990.

[21] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sciences*, Vol.55, No.1, pp.119–139, 1997.

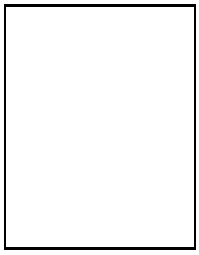
[22] G. Ridgeway, D. Madigan and T. Richardson, "Boosting methodology for regression problems," in: D. Heckerman and J. Whittaker, eds., *Proc. Artificial Intelligence and Statistics*, 1999, pp.152–161.
[23] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in: S. Mendelson and A. Smola, eds., *Advanced Lectures in Machine Learning*. Springer Verlag, 2003, pp.119–184.
[24] R.H. Myers, *Classical and Modern Regression with Applications*. 2nd Edition, Boston: PWS-KENT, 1990.
[25] D.H. Wolpert, "Stacked generalization," *Neural Networks*, Vol.5, No.2, pp.241–259, 1992.
[26] L. Breiman, "Stacked regressions," *Machine Learning*, Vol.24, pp.49–64, 1996.
[27] L.K. Hansen and J. Larsen, "Linear unlearning for cross-validation," *Advances in Computational Mathematics*, Vol.5, pp.269–280, 1996.
[28] G. Monari and G. Dreyfus, "Withdrawing an example from the training set: an analytic estimation of its effect on a non-linear parameterised model," *Neurocomputing*, Vol.35, pp.195–201, 2000.
[29] G. Monari and G. Dreyfus, "Local overfitting control via leverages," *Neural Computation*, Vol.14, pp.1481–1506, 2002.
[30] S. Chen and S.A. Billings, "Representation of non-linear systems: the NARMAX model," *Int. J. Control*, Vol.49, No.3, pp.1013–1032, 1989.
[31] S.A. Billings and S. Chen, "Extended model set, global data and threshold model identification of severely non-linear systems," *Int. J. Control*, Vol.50, No.5, pp.1897–1923, 1989.
[32] S. Chen, X.X. Wang and C.J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," submitted to *IEEE Trans. Systems, Man and Cybernetics, Part B*, 2004.
[33] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, Vol.4, No.3, pp.415–447, 1992.
[34] S.A. Billings, S. Chen and R.J. Backhouse, "The identification of linear and non-linear models of a turbocharged automotive diesel engine," *Mechanical Systems and Signal Processing*, Vol.3, No.2, pp.123–142, 1989.
[35] G.E.P. Box and G.M. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden Day Inc., 1976.



Sheng Chen (SM'97) received the B.Eng. degree in control engineering from the East China Petroleum Institute, Dongying, China, in 1982 and the Ph.D. degree in control engineering from the City University, London, U.K., in 1986.

He joined the School of Electronics and Computer Science, University of Southampton, Southampton, U.K., in September 1999. He previously held research and academic appointments at the University of Sheffield, Sheffield, U.K., the University of Edinburgh, Edinburgh, U.K., and University of Portsmouth, Portsmouth, U.K. His recent research works include adaptive nonlinear signal processing, modeling and identification of nonlinear systems, machine learning and neural networks, finite-precision digital controller design, evolutionary computation methods, and optimization. He has published over 200 research papers.

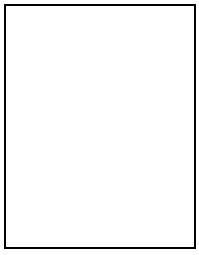
In the database of the world's most highly cited researchers in various disciplines, compiled by Institute for Scientific Information (ISI) of the USA, Dr. Chen is on the list of the highly cited researchers in the category that covers all branches of engineering subjects.



Xia Hong (SM'02) received the B.Sc. and M.Sc. degrees from National University of Defense Technology, Changsha, China in 1984 and 1987, respectively, and the Ph.D. degree from the University of Sheffield, Sheffield, U.K., in 1998, all in automatic control.

She worked as a research assistant in the Beijing Institute of Systems Engineering, Beijing, China, from 1987-1993. She worked as a research fellow in the Department of Electronics and Computer Science, University of Southampton, Southampton, U.K., from 1997-2001. She is currently a lecturer at the Department of Cybernetics, University of Reading, Reading, U.K. She is actively engaged in research into neurofuzzy systems, data modeling and learning theory and their applications. Her research interests include system identification, estimation, neural networks, intelligent data modeling, and control. She has published over 40 research papers, and co-authored a research book.

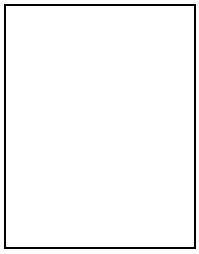
Dr. Hong received a Donald Julius Groen Prize from IMechE, U.K., in 1999.



Chris Harris receiving the B.Sc. degree from the University of Leicester, Leicester, U.K., the M.A. degree from the University of Oxford, Oxford, U.K., and the Ph.D. degree from the University of Southampton, Southampton, U.K.

He previously held appointments at the University of Hull, Hull, U.K., the UMIST, Manchester, U.K., the University of Oxford, Oxford, U.K., and the University of Cranfield, Cranfield, U.K., as well as being employed by the U.K. Ministry of Defense. He returned to the University of Southampton as the Lucas Professor of Aerospace Systems Engineering in 1987 to establish the Advanced Systems Research Group and, more recently, Image, Speech and Intelligent Systems Group. His research interests lie in the general area of intelligent and adaptive systems theory and its application to intelligent autonomous systems such as autonomous vehicles, management infrastructures such as command & control, intelligent control, and estimation of dynamic processes, multi-sensor data fusion, and systems integration. He has authored and co-authored 12 research books and over 300 research papers, and he is the associate editor of numerous international journals.

Dr. Harris was elected to the Royal Academy of Engineering in 1996, was awarded the IEE Senior Achievement medal in 1998 for his work in autonomous systems, and the highest international award in IEE, the IEE Faraday medal, in 2001 for his work in intelligent control and neurofuzzy systems.



Xunxian Wang received his PhD degree in the control theory and application field from Tsinghua University, Beijing, China, in July 1999.

From August 1999 to August 2001, he was a postdoctoral researcher in the State Key laboratory of Intelligent Technology and Systems, Beijing, China. From September 2001, he has been a research associate and now research fellow at the University of Portsmouth, Portsmouth, U.K. Dr. Wang's main interests are in machine learning and neural networks, control theory and systems as well as robotics.